

# Possible Jenkins Umbrella Foundations

- Purpose
- Foundations
  - Apache
  - Eclipse
  - SFC
  - SPI

## Purpose

Jenkins needs an umbrella foundation of some sort, at a minimum to provide a legal home for the Jenkins trademark/IP, a place to properly handle donations, etc. How much more than that is needed is an open question. This page is intended to provide a venue for discussing the pros and cons of each of the possible homes.

When adding a pro or con, or commenting on existing ones, please make sure to leave your name/username.

## Foundations

Foundations are in alphabetical order.

### Apache

- Pros
  - Well know and respected (pablaasmo)
    - Some companies have not contributed to Jenkins because they believe it is controlled by CloudBees. Apache's integrity will address those concerns.
  - Gives good freedom to community to decide framework/design etc.(pablaasmo)
  - real community over code mind (olamy)
  - trademark and legal help possible (olamy)
  - MIT license is compatible with ASL. (abayer)
  - already in place process to sync with maven central repo (olamy : will need only a groupId change)
  - INFRA in place (jira, wiki, maven repos for snapshots) (olamy)
- Cons
  - Release process ? (weekly release will be more complicated due to vote process) (olamy)
    - *The definition of an ASF release (<http://www.apache.org/dev/release.html>) is something that's publicized outside of the project's developers mailing list, so developer snapshots can be released quickly. A proper release requires a 72-hour vote (bdelacretaz)*
    - *The ASF process does not requires a 72-hour vote. It says it "**should** generally be permitted to run for at least 72 hours to provide an opportunity for all concerned persons to participate regardless of their geographic locations."* (<http://www.apache.org/foundation/voting.html>) (elechamny)
  - new committer entry (olamy : IHMO not a real issue as the goal is to probably to move only core and bundled plugins)
    - *Adding new committers is not complicated but requires a 72-hour vote and there's a delay for creating new accounts that's currently up to a week (bdelacretaz)*
  - back to svn (AFAIK asf use a central svn repo) (olamy : btw is it a problem ?) (*andrew: fwiw, ASF has talked about having a git repo, but no one there has stepped up to get it set up/supported - that may be something we'd be able to help with*)
    - *Is this really true? Why could we not continue to run with github? Can someone explain what the infrastructure constraints are? (magnayn)*
      - *I can go into more detail on this based on conversations I've had with Doug Cutting - I'll try to get that summed up somewhere. (abayer)*
    - btw we can ask ASF for jenkins first using a git canonical repository hosted @asf (olamy) (@magnayn btw you can use git svn as svn repos are mirrored to git)
      - git-svn != git. You loose several important capabilities; most importantly you can never merge in git and push back to svn. This is a particular issue when you want to have maintenance branches, and merge back to the development tree. git-svn is useful for situations where you're corporate-locked to SVN, but it's a huge step backwards from git (and github) proper - you might do something in github that's impossible to ever commit back to svn. (magnayn)
    - *Apache project code must be in the ASF's central svn repo, but you can use git to a certain extent, see <http://git.apache.org> - git support might be extended in the future but there are no definite plans at this time as far as I know (bdelacretaz)*
      - *My biggest problem with this is it smells like politics (magnayn). Many of us spend enough time in \$dayjob fighting IT departments dubious ideas without having them on our side projects. Notably:*
        - *There is no such thing as 'canonical' in DVCS terms. I get worried about this requirement as being driven by an infrastructure team demarcation requirement. This solves no problems for Jenkins, and just creates new ones (it's just the same as ORCL deciding when and where the repo can be, not the community). If, however, it could be done such that we proceed as now - where development work is organised around github but the requirement being something like 'the SHA1 of any release must exist in >this< git repository - then that's totally fine.*
    - legal issues regarding gpl libs (olamy : btw same for eclipse)
      - See <http://apache.org/legal/resolved.html> for info about acceptable licenses for ASF project dependencies (bdelacretaz)
  - See Also
    - [How the ASF works](#)
    - [What makes Apache projects different](#)

## Eclipse

- Pros
  - Gets the possibility to join with Hudson again (pablaasmo)
  - Could lead to working closer to the Mylin guys and their Hudson/Jenkins connector (vtintillier)
  - Eclipse projects can use git nowadays (vtintillier)
  - Has a lot of experience to share regarding plugin ecosystems and technology (jutzig)
- Cons
  - EPL is not MIT compatible (magnayn)
  - Might have more restrictions on how the framework should be. OSGI etc (pablaasmo).
  - Their projects websites are crap and I find them hard to find (vtintillier)
  - If our goal is not to re-unite with Hudson, it's unlikely that Jenkins will be accepted (kutzi)
  - The eclipse IP process makes it very hard for occasional contributors. Many will just give up. (magnayn)
    - *For the contributor it's simply a bugzilla and a short statement that the contributor has the right to donate the code. Is that so hard? (jutzig)*
      - *It appears so - which is why if I see a project is @ eclipse, I give up trying to contribute (magnayn) : [The Tragedy of eclipse Eclipse.org JGit follies continue](#)*
  - Bugzilla. Ick. (abayer)
  - Compared with ASF, more decisions in Eclipse depend on the Board and the [Eclipse Staff](#). The bylaws are 24 pages (see [here](#) and the rest of the [Governance Documents](#)). As a specific, possibly infrequent, decision, see [this comment](#) from the Foundation on community reactions to the donation (pelegri)

## SFC

<http://sfconservancy.org/>

- Pros
  - "takes care of the projects' needs that do not relate directly to software development" (magnayn)
  - Equivalent to being a non-profit organisation, without having to form & maintain one (magnayn)
  - Can hold assets like (c), TM
  - Legal protection & enforcement
- Cons
  - No infrastructure support. (abayer)

## SPI

[www.spi-inc.org](http://www.spi-inc.org)

- Pros
  - Apparently very liberal regarding which license to use (kutzi)
- Cons
  - No infrastructure support. (abayer)